

Inequity Aversion Pricing over Social Networks: Approximation Algorithms and Hardness Results^{*}

Georgios Amanatidis[†] Evangelos Markakis[†] Krzysztof Sornat[‡]

June 22, 2016

Abstract

We study a revenue maximization problem in the context of social networks. Namely, we consider a model introduced by Alon, Mansour, and Tennenholtz (EC 2013) that captures *inequity aversion*, i.e., prices offered to neighboring vertices should not be significantly different. We first provide approximation algorithms for a natural class of instances, referred to as the class of single-value revenue functions. Our results improve on the current state of the art, especially when the number of distinct prices is small. This applies, for example, to settings where the seller will only consider a fixed number of discount types or special offers. We then resolve one of the open questions posed in Alon et al., by establishing APX-hardness for the problem. Surprisingly, we further show that the problem is NP-complete even when the price differences are allowed to be relatively large. Finally, we also provide some extensions of the model of Alon et al., regarding either the allowed set of prices, or the demand type of the clients.

1 Introduction

We study a differential pricing optimization problem in the presence of network effects. Differential pricing is a well known practice in everyday life and refers to offering a different price to potential customers for the same service or good. Examples include offering cheaper prices when launching a new product, making special offers to gold and silver members of an airline miles program, offering discounts at stores during selected periods, and several others.

We are interested in studying differential pricing in the context of a social network. Imagine a network connecting individuals (who are seen as potential clients here) with their friends, family, or colleagues, i.e., with people who can exert some influence on them. One can have in mind other forms of abstract networks as well, e.g., a node could represent a geographic region, a neighborhood within a city, a type of profession, a social class, and edges can represent interactions or proximity. The presence of such a network creates *externality effects*, meaning that the decision of a node to acquire a new product or a new service is affected by the fact that some other nodes within her social circle (her neighborhood in the graph) already did so. A typical example of positive externalities is when someone becomes more likely to buy a new product due

^{*}A preliminary conference version appeared in MFCS 2016 [3].

[†]Athens University of Economics and Business, Athens, Greece. Emails: {gamana, markakis}@aueb.gr

[‡]University of Wrocław, Wrocław, Poland. Email: krzysztof.sornat@cs.uni.wroc.pl

to the positive reviews by a friend who already bought it in the past. Modeling positive externalities has led to a series of works that study marketing strategies for maximizing the diffusion of a new product, [10, 17], or the total revenue achieved, [16] (see also the Related Work section).

However, there also exist negative externality effects that can arise in a network. One example is the purchase of a product with the intention to show off and be a locally unique owner, e.g., a new type of expensive car, or clothes (also referred to as *invidious consumption*, see [6]). In such a case, a node may be deterred from buying the same product, if a neighboring node already did so. A second example of negative externalities, which is the focus of our work, and arises from differential pricing, is *inequity aversion*, see e.g., [5] and [11]. This simply means that a customer may experience dissatisfaction if she realizes that other people within her social circle, were offered a better deal for the same service. Hence, significant price differences, can create a negative response of some customers towards a product. Inequity aversion can also arise under a different, but equally applicable, interpretation: nodes may correspond to retail stores and an edge can signify proximity, so that clients could choose among these stores. Again, having significantly different prices to the same products is not desirable.

To capture the need for avoiding such phenomena, the relatively recent work of [2] introduced a model for pricing nodes over a social network. The main idea is to impose constraints on each edge, specifying that the price difference between two neighbors should be bounded by some (endogenous) parameter, determined by the two neighbors. On top of this, the seller is also allowed to not make a price offer to some nodes (referred to as introducing *discontinuities*, see the related discussion in Section 2), in which case the difference constraints do not apply for the edges incident to these nodes. Assuming a finite set of available prices, unit-demand users, and digital goods (i.e., the supply can cover all the demand) the problem is to find a feasible price vector that satisfies the edge constraints and maximizes the total revenue. In its more general form the problem was shown to be NP-complete, but exact or approximation algorithms were derived for some interesting cases.

Contribution: We revisit the model introduced by [2] (namely Model II of [2], which is the more general one), and study the approximability of the underlying revenue maximization problem. We resolve one of the open questions posed in [2], regarding the complexity of the problem under the natural class of the so-called *single-value* revenue functions. Simply put, this means that the revenue extracted by each node is exactly the price offered to her, as long as the price does not exceed her valuation for the product (the usual assumption made in auction settings as well). We first establish APX-hardness for this class answering the question of [2], and we also show that the problem is NP-complete even when the price differences are allowed to be relatively large (a case that could be thought easier to handle). We then provide approximation algorithms that improve some of the currently known results. Our improvement is stronger when the number of distinct prices is small. This applies for example to many settings where the seller will only consider a fixed number of discount types or special offers to selected customers. As the number of available price offers becomes large, the performance of our algorithm degrades to a logarithmic approximation. Finally, we also provide 2 extensions of these results; the first concerns a more general model where the allowed prices come from a set of k arbitrary integers, instead of using price sets of the form $\{1, 2, \dots, k\}$, as done in [2], and the second concerns a multi-unit demand setting (see Subsection 4.3 and Section 6).

Related Work: Price discrimination is well studied in various domains in economics and is also being applied to numerous real life scenarios. The algorithmic problem of differential pricing over

social networks is a more recent topic, initiated by [16]. The work of [16] studied a model with positive externalities, where the valuation of a player may increase as more friends acquire a good, and analyzed the performance of a very intuitive class of pricing strategies. Further improvements on the performance of such strategies were obtained later on by [12]. The work of [1] also considers a pricing problem but in an iterative fashion, where the seller is allowed to reprice a good in future rounds. Revenue maximization under a mechanism design approach was also taken in [15] under positive network externalities. Finally, positive externalities have been used to model the diffusion of products on a network, see, among others, the exposition in [19].

Negative externalities within networks, as we focus on here, are less studied in the literature. For the concept of inequity aversion, see e.g., [5, 11]. The work most closely related to ours is [2], which introduced the model that we consider here. Efficient algorithms were obtained for the case where discontinuities are not allowed (even for more general revenue functions), and also for networks with bounded treewidth. An approximation ratio of $1/(\Delta + 1)$ was also provided, where Δ is the maximum degree. Similar results were shown for a stochastic version of the model. Finally, other types of negative externalities have been considered e.g., in [4, 6] which study the effects of invidious consumption.

2 Definitions and Preliminaries

The social network is represented as an undirected graph $G = (V, E)$, with $|V| = n$. We assume that a provider of some good or service has a finite set P of available prices that he could offer to the nodes. In most of our presentation, we assume as in [2], that the available prices are given by $P = \{1, 2, \dots, k\}$. In Subsection 4.3, we show how to extend the analysis when P is an arbitrary set of k positive integers, i.e., $P = \{p_1, p_2, \dots, p_k\}$.

We assume every node has a unit-demand for the same product and that the supply of the seller is enough to cover the demand of all nodes. For every node $v \in V$, we associate a revenue function $R_v : \{1, 2, \dots, k\} \rightarrow \mathbb{N}$ that maps an offered price p_v to the revenue that the provider gains from this offer. In this paper, we focus on a simple and intuitive class of revenue functions, also studied in [2]. In particular, for a node $v \in V$, R_v is called a *single value revenue function*, if there exists a value $val(v)$ such that when offered a price p_v :

$$R_v(p_v) = \begin{cases} p_v & \text{if } val(v) \geq p_v \\ 0 & \text{if } val(v) < p_v \end{cases}$$

We assume from now on that every node has a single value revenue function. We also assume that $val(v) \in P$, for every $v \in V$. This is because for revenue maximization, that we are interested in, nodes with $val(v) > k$, can only yield a revenue of k , and could be replaced by $val(v) = k$, i.e., the highest possible price. Also for values that are less than k , and not integers, we can again extract only an integer revenue, given the form of P . Finally, any node v with $val(v) < 1$ can be deleted without affecting the optimal revenue (see the concept of *discontinuity* defined below), so we can completely ignore such nodes to begin with. Thus, we consider only instances with $val(v) \in \{1, 2, \dots, k\}, \forall v \in V$.

Given a vector $\mathbf{p} = (p_v)_{v \in V}$ of prices offered to the nodes, the total revenue is $R(\mathbf{p}) = \sum_{v \in V} R_v(p_v)$. Hence, our goal is to find a price vector that maximizes the total revenue. At the same time, however, we want to capture the effect of *inequity aversion* [5, 11] in social networks. This means that a node may experience dissatisfaction if she sees that other nodes within her social circle, were

offered a better deal for the same service. Hence, significant price differences, create negative externalities among users.

To avoid such situations the model introduced in [2] has constraints on each edge, stating that the price difference between two neighbors u, v is bounded, i.e., $p_u - p_v \leq \alpha(u, v)$ and $p_v - p_u \leq \alpha(v, u)$, for every $(u, v) \in E$. Here, $\alpha(\cdot, \cdot) \geq 0$ is integer-valued (given that the prices are also integers) and note that in general is non-symmetric. Furthermore, the seller is also allowed not to make an offer to certain nodes. Formally, this is captured by having one more price option, which we denote by \perp , with $R_v(\perp) = 0$. Setting $p_v = \perp$ to a node, means that the provider does not make any offer to v , and there is no price restriction on the edges that are incident to v . We can essentially think about this as deleting these vertices from the graph. We will refer to setting $p_v = \perp$ to a node $v \in V$, as introducing a *discontinuity* on v . Avoiding making an offer can be thought of as choosing not to promote a product or service within a certain region or within a certain social group. In terms of optimization, allowing discontinuities can help the seller in producing much higher revenue (than without discontinuities) as Proposition 3.3 in Section 3 states.

Given this model, the set of feasible price vectors is then: $\mathcal{F} = \{\mathbf{p} : \forall v \in V, p_v \in P \cup \{\perp\}, \text{ and } \forall (u, v) \in E, p_u \neq \perp \wedge p_v \neq \perp \Rightarrow p_u - p_v \leq \alpha(u, v) \wedge p_v - p_u \leq \alpha(v, u)\}$. Therefore, the problem we study is:

Inequity Aversion Pricing: Given a graph with edge constraints, and a single-value revenue function for each node, find a feasible price vector that maximizes the total revenue, i.e., find $\mathbf{p} \in \mathcal{F}$ that achieves $\max_{\mathbf{p} \in \mathcal{F}} \sum_{v \in V} R_v(p_v)$.

Some cases of this problem, as well as the variant where no discontinuities are allowed, are already known to be polynomial time solvable [2]. Regarding hardness, although the problem is NP-hard for more general revenue functions, it was posed as an open question whether NP-hardness still holds for single value revenue functions (the hardness result in [2] requires instances with revenue functions that cannot be captured by single value ones).

3 Warmup: Basic Facts and Single-price Solutions

In this section, we present a simple algorithm and some basic observations, which we use later on, in Section 4.

Let $v_{\max} = \max_{v \in V} \text{val}(v) \leq k$, and $\text{MAX} = \sum_{v \in V} \text{val}(v)$. Given an instance of the problem, we denote by OPT the revenue of an optimal solution. The quantity MAX is clearly an upper bound on the optimal revenue, hence $\text{OPT} \leq \text{MAX}$.

We will refer to a solution as being a *single-price* solution, if it charges the same price to every node without introducing discontinuities. Note that this is always a feasible solution since all the edge constraints are satisfied. The revenue extracted by a single-price algorithm that uses the price of p for all nodes is equal to $p \cdot |\{v \in V : \text{val}(v) \geq p\}|$.

To understand whether single-price solution can be of any help for our setting, we can examine the performance of the best possible single price. The following observation suggests that we do not need to try too many values, even if v_{\max} is very large.

Lemma 3.1. *In order to find the optimal single-price solution, it suffices to check at most $\min\{n, v_{\max}\}$ possible prices.*

Proof. There are at most $\min\{n, v_{\max}\}$ different values in the set $\{\text{val}(v) : v \in V\}$. It is never optimal to use any price $p \notin \{\text{val}(v) : v \in V\}$. Indeed, if $p \in (\text{val}(v_1), \text{val}(v_2))$, where $\text{val}(v_1)$ and

$val(v_2)$ are two consecutive distinct values for some nodes $v_1, v_2 \in V$, then it is strictly better to set the price to $val(v_2)$. For the same reason, it is suboptimal to set a price that is less than the minimum value across nodes, while if we use a price $p > v_{\max}$ then we gain no revenue. \square

Hence in $O(\min\{n, v_{\max}\})$ steps, we can select the best single-price solution. Let us denote by R_{SP} the revenue raised by this solution. The performance of R_{SP} has been analyzed in a different context¹ by [14], where it was shown that it achieves a $\Theta(\ln n)$ -approximation. Here we give a slightly tighter statement, which we utilize in later sections for small values of v_{\max} .

Theorem 3.2. *For any number n of agents, the optimal single-price solution achieves a $1/H_r$ -approximation, where $r = \min\{n, v_{\max}\}$, and H_ℓ is the ℓ -th harmonic number, i.e.,*

$$R_{\text{SP}} \geq \frac{\text{MAX}}{H_r} \geq \frac{\text{OPT}}{H_r}.$$

Furthermore, the approximation guarantee is tight.

The proof follows from the proof of Theorem 4.5 in Section 4. One interesting point here, is that single-price solutions do not use any discontinuities. If R_{ND} is the maximum revenue without using any discontinuities, clearly $R_{\text{ND}} \geq R_{\text{SP}}$. And as we mentioned in Section 2, it is possible to find the optimal solution that does not use discontinuities in polynomial time; so why use something worse instead of R_{ND} ? Actually, besides being harder to argue about, R_{ND} turns out to be as bad an approximation as R_{SP} , in the worst case. Hence, the proposition below reveals that introducing discontinuities can cause a significant increase in the optimal revenue achievable by the seller, compared to what can be achieved without discontinuities.

Proposition 3.3. *The optimal solution with no discontinuities achieves a $1/H_r$ -approximation, where $r = \min\{n, v_{\max}\}$, and this approximation guarantee is tight.*

Proof. The approximation guarantee follows from the fact that single-price solutions do not use any discontinuities. To see that without using any discontinuities one cannot always do better, we can modify slightly the examples that give tightness in the proof of Theorem 4.5.

In each case, connect a new vertex v with value 1 to every vertex u and set $\alpha(u, v) = \alpha(v, u) = 0$. The optimal solution is to put a discontinuity on v and maximize the revenue of every other vertex. When discontinuities are not allowed though, a solution cannot do better than R_{SP} , since all prices have to be equal in such a solution. It is easy to see that we still get the ratios of the proof of Theorem 4.5, namely $\frac{1}{H_n}$ and $\frac{n+1}{n \cdot H_{v_{\max}}}$. \square

4 Approximation of Inequity Aversion Pricing

In this section we present new approximation algorithms for the problem by exploiting ways in which setting discontinuities in certain nodes can help. Our main result is an approximation algorithm, with a ratio of $(H_k - 0.25)^{-1}$. Even though asymptotically this is no better than the optimal single-price algorithm, it does yield better ratios for instances where k is a small constant. The motivation for studying cases where the set of available prices is a small constant is that a seller may be willing to offer only specific types of discount to selected customers, e.g., 20% or 30% off the regular price and so on, rather than using an arbitrary set of prices.

We start below with the case of $k = 2$, before we move to the more general case.

¹The work of [14] studied an auction pricing problem without the presence of social networks.

4.1 A 0.8-approximation Algorithm when $P = \{1, 2\}$ via Vertex Cover

In this subsection, we assume the available prices are 1, 2, or \perp . Despite this restriction, the problem still remains non-trivial, and it is currently not known if it is NP-complete or not. Given the discussion in Section 2, we will also assume that for every node $v \in V$, $val(v) \in \{1, 2\}$. For such instances we already have a $\frac{2}{3}$ -approximation by Theorem 3.2, that does not use discontinuities. The difficulty in improving this factor is in finding a way of selecting appropriate nodes to set to \perp .

Before we describe our algorithm, let us illustrate the main idea. Consider an instance of the problem on a graph $G = (V, E)$. Suppose we plan to find a feasible price vector, such that for each u , either $p_u = \perp$ or $p_u = val(u)$. Since the possible prices are only 1 and 2, if $val(u) = 1$, then for any $(u, v) \in E$, $\alpha(u, v)$ is not restrictive, while if $val(u) = 2$, then for any $(u, v) \in E$, $\alpha(u, v)$ is restrictive only if $\alpha(u, v) = 0$ and $val(v) = 1$. So, we could remove any edge except from edges in $E' = \{(u, v) \in E : val(u) = 2, val(v) = 1, \alpha(u, v) = 0\}$. Note that this defines a bipartite subgraph $G' = (V_1, V_2, E')$ of G , where $V_i = \{v \in V : val(v) = i\}$. Since this new instance has less restrictions, the optimal revenue OPT' is at least as good as the optimal revenue OPT of the original instance.

Consider a vertex cover S in G' . The crucial observation is that we can satisfy all the edge constraints regarding edges between V_1 and V_2 , by introducing discontinuities on the vertices of S . Since S covers all the edges between V_1 and V_2 , the edge constraints between V_1 and V_2 in the original graph G are now non-existent. If we also set a price of 1 on the remaining vertices of V_1 and a price of 2 on the remaining vertices of V_2 , all the original constraints are satisfied. Thus, we have constructed a feasible solution for G .

The revenue of such a solution is $MAX - val(S)$, where $MAX = \sum_{v \in V} val(v) = |V_1| + 2 \cdot |V_2|$ and $val(S) = \sum_{v \in S} val(v)$. Hence, the best outcome of such an algorithm is achieved when S is a minimum weighted vertex cover (using the values as weights) rather than just any vertex cover. For the analysis however, it suffices to compute just a minimum vertex cover (see the Remark after the proof of Theorem 4.1). Moreover, by the Kőnig–Egerváry Theorem, we can compute this in polynomial time for bipartite graphs (e.g., see [20]).

Finally, the algorithm compares the best of two outcomes, the solution outlined above and the solution discussed in Section 3. Hence, we define $ALG = \max\{R_{SP}, MAX - val(S)\}$, where R_{SP} is the maximum revenue achieved by setting a fixed price to every node.

Algorithm 1: A 0.8-approximation when $P = \{1, 2\}$

- 1 Given the graph $G = (V, E)$, construct the bipartite graph $G' = (V_1, V_2, E')$ with $V_i = \{v \in V : val(v) = i\}$ and $E' = \{(u, v) \in E : val(u) = 2, val(v) = 1, \alpha(u, v) = 0\}$
 - 2 Find a minimum vertex cover on G' , say $S \subseteq V$
 - 3 Set \perp to all vertices of S
 - 4 Set a price of 1 to every $v \in V_1 \setminus S$ and a price of 2 to every $v \in V_2 \setminus S$. Let R^* be the revenue obtained by this solution
 - 5 Compute the optimal single-price solution, as described in Section 3, with revenue R_{SP}
 - 6 Return the solution that achieves $\max\{R^*, R_{SP}\}$
-

Theorem 4.1. *Algorithm 1 achieves a 0.8-approximation for the Inequity Aversion Pricing problem when $P = \{1, 2\}$. Furthermore, this ratio is tight.*

Proof. Let ALG denote the revenue obtained by Algorithm 1 and let β be its approximation ratio that we attempt to determine. Assume that $\beta < 0.8$. Then there exists some $\varepsilon > 0$ such that $\beta = 0.8 - \varepsilon$. To arrive at a contradiction, we are going to show that $\beta \geq \gamma = 0.8 - \varepsilon/2$.

We will distinguish some cases, depending on the value of ALG . First of all, note that if $\text{ALG} \geq \gamma \cdot \text{MAX}$, then we trivially obtain a γ -approximation: $\frac{\text{ALG}}{\text{OPT}} \geq \frac{\gamma \cdot \text{MAX}}{\text{MAX}} \geq \gamma$. From now on, assume that $\text{ALG} < \gamma \cdot \text{MAX}$. The following turns out to be a very useful upper bound for OPT .

Claim 4.2. *Let S denote a minimum vertex cover in the graph G' (defined in step 1 of Algorithm 1). Then, $\text{OPT} \leq \text{OPT}' \leq \text{MAX} - |S|$.*

Proof of Claim 4.2. The first inequality is straightforward (see also the discussion before the theorem). For the second inequality, note that by the König–Egerváry Theorem, the maximum matching in G' has the same cardinality as S . Let M be such a maximum matching. By the definition of G' , for each edge $(u, v) \in M$ the nodes u and v have different values, say $\text{val}(u) = 2$ and $\text{val}(v) = 1$. Because of (u, v) , an optimal solution must lose at least one unit of revenue in comparison with MAX . Indeed, since $\alpha(u, v) = 0$, an optimal solution either sets a discontinuity on one of these two nodes, or it sets the same price. If this common price is 1, we lose one unit from node v , whereas if it is 2 we do not extract revenue from u . The claim follows. \triangleleft

We know that $\text{ALG} = \text{MAX} - \text{val}(S)$ and also $\text{val}(S) \leq 2|S|$. Thus, $|S| \geq \frac{1}{2}(\text{MAX} - \text{ALG})$. If we combine this with Claim 4.2, we have

$$\text{OPT} \leq \frac{1}{2}(\text{MAX} + \text{ALG}). \quad (1)$$

To proceed with the analysis, we divide the interval $[0, \gamma \cdot \text{MAX}]$ into smaller subintervals of the form $[\frac{i-1}{m} \cdot \gamma \cdot \text{MAX}, \frac{i}{m} \cdot \gamma \cdot \text{MAX}]$ for some fixed large m and $i \in \{1, \dots, m\}$. Notice that m is just a parameter in the analysis and has nothing to do with the input. We consider cases depending on where exactly the value of ALG falls. In particular, let i^* be the following interval index: $i^* = \left\lceil \frac{m+2}{2-\gamma} \right\rceil$.

Case (i): $\text{ALG} \in [\frac{i-1}{m} \cdot \gamma \cdot \text{MAX}, \frac{i}{m} \cdot \gamma \cdot \text{MAX}]$ with $i \geq i^*$.

Using inequality (1), we have:

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{\frac{i-1}{m} \cdot \gamma \cdot \text{MAX}}{\frac{1}{2}(\text{MAX} + \text{ALG})} \geq \frac{\frac{i-1}{m} \cdot \gamma \cdot \text{MAX}}{\frac{1}{2}(\text{MAX} + \frac{i}{m} \cdot \gamma \cdot \text{MAX})} = \frac{\frac{i-1}{m} \cdot \gamma}{\frac{1}{2}(1 + \frac{i}{m} \cdot \gamma)}.$$

In order to ensure a γ -approximation, it suffices to have

$$\frac{\frac{i-1}{m} \cdot \gamma}{\frac{1}{2}(1 + \frac{i}{m} \cdot \gamma)} \geq \gamma \iff \frac{i-1}{m} \geq \frac{1}{2} \left(1 + \frac{i}{m} \cdot \gamma \right) \iff i \geq \frac{m+2}{2-\gamma}.$$

But this last inequality holds since $i \geq i^*$. Therefore, in this case, the algorithm achieves a γ -approximation.

Case (ii): $\text{ALG} < \frac{i^*-1}{m} \cdot \gamma \cdot \text{MAX}$.

Again, we use inequality (1), but now the lower bound of ALG comes from Theorem 3.2, which gives a guarantee for the optimal single-price solution:

$$\frac{\text{ALG}}{\text{OPT}} \geq \frac{R_{\text{SP}}}{\frac{1}{2}(\text{MAX} + \text{ALG})} \geq \frac{\frac{1}{H_2} \text{MAX}}{\frac{1}{2} \text{MAX} (1 + \gamma \cdot \frac{i^*-1}{m})} = \frac{4/3}{1 + \gamma \cdot \frac{i^*-1}{m}}.$$

Like in case (i), it suffices to have

$$\frac{4/3}{1 + \gamma \cdot \frac{i^*-1}{m}} \geq \gamma \iff 4 \geq 3\gamma \left(1 + \gamma \cdot \frac{i^*-1}{m} \right).$$

Using an obvious upper bound for i^* , it suffices for γ to satisfy the following:

$$4 \geq 3\gamma + 3\gamma^2 \cdot \frac{\frac{m+2}{2-\gamma} + 1 - 1}{m} \iff \frac{6}{m}\gamma^2 + 10\gamma - 8 \leq 0.$$

Clearly, there is some $m^* \in \mathbb{N}$, such that

$$\frac{6}{m^*}(0.8 - \varepsilon/2)^2 + 10(0.8 - \varepsilon/2) - 8 \leq 0.$$

Thus, the approximation ratio β of Algorithm 1 is at least $0.8 - \varepsilon/2$, which contradicts the choice of ε . Hence, $\beta \geq 0.8$.

To see why the ratio of the algorithm is tight, we can construct an infinite family of examples as follows: Consider a graph of 4 nodes $\{v_1, v_2, v_3, v_4\}$ such that $val(v_1) = val(v_2) = 2$, and $val(v_3) = val(v_4) = 1$. There are only two edges, namely (v_2, v_3) and (v_2, v_4) . Suppose $\alpha(\cdot, \cdot) = 0$. The optimal revenue here is 5 by offering a price of 1 to v_2, v_3, v_4 and a price of 2 to v_1 . On the other hand, the optimal single-price algorithm achieves a revenue of 4, either with a price of 1 or 2. Also, a minimum (weighted or not) vertex cover here is either $\{v_2\}$ or $\{v_3, v_4\}$. In both cases, the revenue by setting \perp to the vertex cover is 4. We can add many copies of this construction (and possibly some extra edges with $\alpha(e) \geq 1$ for a connected example) to turn this into an infinite family of tight examples. For an illustration, see Figure 1. \square

Remark 1. It seems appealing to try to exploit the fact that we can solve the minimum weighted vertex cover problem in polynomial time for bipartite graphs. However, as our analysis shows, using the weighted version of vertex cover, instead of the unweighted one, does not yield any better approximation.

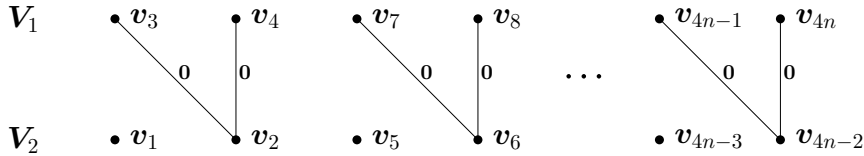


Figure 1: Algorithm 1 is tight on such instances. Only the relevant edges are shown.

4.2 An Approximation Algorithm for $k > 2$

We now consider the case where there are more than two available prices. In order to improve the approximation guarantee of Theorem 3.2, we reduce the problem to the case of $k = 2$, and use the results of the previous subsection.

Consider an instance of the problem, with available prices in $\{\perp, 1, 2, \dots, k\}$. As discussed in Section 2, we may assume that $val(v) \in \{1, 2, \dots, k\}$ for every $v \in V$. We create another instance, where we set the value of every node with $val(v) > 1$ to be equal to 2. We can then run Algorithm 1 from Subsection 4.1 on this new instance. At the same time, we can also compute the optimal single-price solution for the original instance, and pick the best among these two solutions. This yields Algorithm 2, described below.

Algorithm 2: An algorithm for $k > 2$

- 1 Given an instance I , construct a new instance I' , where for every $v \in V$, $val'(v) = \min\{val(v), 2\}$; everything else remains unchanged
 - 2 Run Algorithm 1 from Subsection 4.1 on instance I' , and let R_* be the revenue obtained
 - 3 Compute the optimal single-price solution without discontinuities, on the original instance I , as described in Section 3, with revenue R_{SP}
 - 4 Return the solution that achieves $\max\{R_*, R_{SP}\}$
-

Clearly, Algorithm 2 runs in polynomial time. Note that the solution returned by the algorithm is feasible. Any single-price solution is always feasible, while Algorithm 1 will produce a price vector that is feasible for I' , and therefore for I , since the edge restrictions in the two instances are the same. Even though asymptotically, this is still a logarithmic approximation, the algorithm achieves significantly better results for small values of k .

Theorem 4.3. *Algorithm 2 achieves a $\frac{1}{H_{v_{\max}} - 0.25}$ -approximation ratio for Inequity Aversion Pricing when the available prices are $\{\perp, 1, 2, \dots, k\}$, with $k \geq 2$.*

Proof. The proof is by induction on v_{\max} . For $v_{\max} = 2$ the result follows from Theorem 4.1 since $0.8 = \frac{1}{H_2 - 0.25}$.

Now assume we have an instance I where $v_{\max} = j > 2$. As usual, let OPT denote the optimal revenue for I and ALG the revenue returned by Algorithm 2. Also, let R_j be the revenue extracted by setting price j at every node, and $V_j = \{v \in V : val(v) = j\}$. We consider two cases.

Case (i): $|V_j| \geq \frac{1}{(H_j - 0.25)j} \cdot OPT$. Then, $\frac{ALG}{OPT} \geq \frac{R_j}{OPT} = \frac{j \cdot |V_j|}{OPT} \geq \frac{\frac{1}{H_j - 0.25} \cdot OPT}{OPT} = \frac{1}{H_j - 0.25}$.

Case (ii): $|V_j| < \frac{1}{(H_j - 0.25)j} \cdot OPT$. Let I^* be an instance derived from I by setting $val^*(v) = \min\{val(v), j - 1\}$, i.e., we only reduce the valuation of the nodes with $val(v) = v_{\max}$ by 1. Let OPT^* denote the optimal revenue for I^* , and ALG^* the revenue returned by Algorithm 2. By the inductive hypothesis we have $ALG^* \geq \frac{1}{H_{j-1} - 0.25} \cdot OPT^*$.

Furthermore, notice that the set of vertices with valuation greater than 1 is the same in both instances. So, Algorithm 2 on input I^* considers exactly the same price vectors as it does on input I , with the exception of the single-price solution that universally uses j . We conclude that $ALG^* \leq ALG$. Next, we prove the following useful claim.

Claim 4.4. $OPT^* \geq OPT - |V_j|$.

Proof of Claim 4.4. Let \mathbf{p} be an optimal price vector for I . Construct the price vector \mathbf{p}^* by decreasing any price that is at least j to $j - 1$. It is straightforward to see that in instance I we have $R(\mathbf{p}^*) \geq R(\mathbf{p}) - |V_j| = OPT - |V_j|$, while in both instances $R(\mathbf{p}^*)$ is the same. What is left to show is that \mathbf{p}^* is feasible for I^* . Observe, however, that the two instances have exactly the same edge restrictions and that, by its definition, \mathbf{p}^* did not increase the price difference between any two vertices compared to \mathbf{p} . Thus, $OPT^* \geq R(\mathbf{p}^*) \geq OPT - |V_j|$. \triangleleft

Now, we can write

$$\frac{ALG}{OPT} \geq \frac{ALG^*}{OPT} \geq \frac{\frac{1}{H_{j-1} - 0.25} \cdot OPT^*}{OPT} \geq \frac{\frac{1}{H_{j-1} - 0.25} \cdot (OPT - |V_j|)}{OPT}$$

$$\begin{aligned}
&\geq \frac{1}{H_{j-1} - 0.25} \left(1 - \frac{\frac{1}{j(H_j - 0.25)} \cdot \text{OPT}}{\text{OPT}} \right) = \frac{1}{H_{j-1} - 0.25} \cdot \frac{j H_j - 0.25 j - 1}{j(H_j - 0.25)} \\
&= \frac{1}{H_{j-1} - 0.25} \cdot \frac{j(H_{j-1} - 0.25)}{j(H_j - 0.25)} = \frac{1}{H_j - 0.25},
\end{aligned}$$

which concludes the proof. \square

4.3 Approximation Algorithms for General Price Sets

We end Section 4 by extending our results when P is an arbitrary set of k positive integers, i.e., $P = \{p_1, p_2, \dots, p_k\}$. This can be seen as a more realistic model, especially for small values of k . In such a case, one could try to directly apply Theorems 3.2, 4.1, or 4.3 for $P' = \{1, 2, 3, \dots, p_k\}$. However, this may produce a very poor approximation when k is small but p_k is large, and feasibility is not guaranteed either. In what follows, P_j denotes $\sum_{i=1}^j \frac{p_i - p_{i-1}}{p_i}$, where $p_0 = 0$.

We begin with a generalization of Theorem 3.2.

Theorem 4.5. *For any number n of agents and possible prices $p_1 < p_2 < \dots < p_k$ the optimal single-price algorithm achieves a ρ -approximation, where $\rho = 1 / \min\{H_n, P_k\}$, i.e.,*

$$R_{\text{SP}} \geq \frac{\text{MAX}}{\min\{H_n, P_k\}} \geq \frac{\text{OPT}}{\min\{H_n, P_k\}},$$

and this approximation guarantee is tight.

Proof. From Lemma 3.1 we know that we only need to consider at most $\min\{n, k\}$ possible values that correspond to the distinct values of the nodes. Let a_j be the number of vertices with value p_j and R_i be the revenue obtained by setting the price of all nodes equal to p_i , i.e.,

$$a_j = |\{v \in V : \text{val}(v) = p_j\}| \text{ and } R_i = \sum_{v \in V} R_v(i) = p_i \cdot \sum_{j=i}^n a_j.$$

Recall that $P_k = \sum_{i=1}^k \frac{p_i - p_{i-1}}{p_i}$, where $p_0 = 0$. Let R_{SP} be the revenue achieved by the optimal one-price algorithm. Then $R_i \leq R_{\text{SP}}$, and we have

$$\text{MAX} = \sum_{v \in V} \text{val}(v) = \sum_{i=1}^k \left((p_i - p_{i-1}) \cdot \sum_{j=i}^n a_j \right) = \sum_{i=1}^k \frac{(p_i - p_{i-1}) \cdot R_i}{p_i} \leq R_{\text{SP}} \cdot P_k.$$

So, we obtain

$$R_{\text{SP}} \geq \frac{\text{MAX}}{P_k} \geq \frac{\text{OPT}}{P_k}. \quad (2)$$

Let us now sort the vertices from V with respect to $\text{val}(v)$ in ascending order, say v_1, \dots, v_n . Let $R_{(i)}$ be the revenue obtained from the vertices $\{v_i, v_{i+1}, \dots, v_n\}$ by setting to all of them $\text{val}(v_i)$ as a price, i.e.,

$$R_{(i)} = \sum_{v \in \{v_i, v_{i+1}, \dots, v_n\}} R_v(\text{val}(v_i)) = (n - i + 1) \cdot \text{val}(v_i).$$

Clearly, $R_{(i)} \leq R_{\text{SP}}$ and we have

$$\text{MAX} = \sum_{i=1}^n \text{val}(v_i) = \sum_{i=1}^n \frac{R_{(i)}}{n - i + 1} \leq R_{\text{SP}} \cdot \sum_{i=1}^n \frac{1}{n - i + 1} = R_{\text{SP}} \cdot \sum_{i=1}^n \frac{1}{i} = R_{\text{SP}} \cdot H_n.$$

Hence, we obtain

$$R_{\text{SP}} \geq \frac{\text{MAX}}{H_n} \geq \frac{\text{OPT}}{H_n}. \quad (3)$$

Putting inequalities (2) and (3) together completes the proof.

To see that this is tight, consider the following family of graphs. For any n take $G(n)$ to be a clique on $\{v_1, v_2, \dots, v_n\}$ and let $\text{val}(v_i) = \frac{n!}{i}$ and $\alpha(u, v) = k = n!$ for every edge. Then

$$\text{OPT} = \text{MAX} = \sum_{i=1}^n \frac{n!}{i} = n! H_n \text{ and } R_{\frac{n!}{i}} = \sum_{j=1}^i \frac{n!}{j} = n! \quad \forall i \in \{1, 2, \dots, n\}.$$

Therefore

$$\frac{R_{\text{SP}}}{\text{OPT}} = \frac{\max_{i \in \{1, 2, \dots, n\}} R_{\frac{n!}{i}}}{n! H_n} = \frac{1}{H_n}.$$

In fact, tightness holds even when $P_k \leq H_n$. Consider an instance where $p_i = i, \forall i \in k$ and $n = k!$. Define $G(k)$ to be a clique on $\bigcup_{i=1}^k V_i$, where $V_i = \{v \in G(k) : \text{val}(v) = i\}$ and

$$\forall i \in \{1, 2, \dots, k-1\} \quad |V_i| = \frac{k!}{i(i+1)}, \text{ and } |V_k| = \frac{k!}{k}.$$

Like before, $\alpha(u, v) = k$ for every edge. It is easy to verify that $\sum_{i=1}^k |V_i| = n$. Then

$$\text{OPT} = \text{MAX} = \sum_{i=1}^k i \cdot |V_i| = k \cdot \frac{n}{k} + \sum_{i=1}^{k-1} i \cdot \frac{n}{i(i+1)} = n \cdot H_k = n \cdot P_k,$$

while

$$\begin{aligned} \forall i \in \{1, 2, \dots, k\} \quad R_i &= \sum_{j=i}^k i \cdot |V_j| = i \cdot n \cdot \left(\frac{1}{k} + \sum_{j=i}^{k-1} \frac{1}{j(j+1)} \right) \\ &= i \cdot n \cdot \left(\frac{1}{k} + \sum_{j=i}^{k-1} \left(\frac{1}{j} - \frac{1}{j+1} \right) \right) = i \cdot n \cdot \frac{1}{i} = n. \end{aligned}$$

Therefore,

$$\frac{R_{\text{SP}}}{\text{OPT}} = \frac{\max_{i \in \{1, 2, \dots, k\}} R_i}{n \cdot P_k} = \frac{1}{P_k}. \quad \square$$

For $k = 2$, Theorem 4.5 yields an approximation ratio of $\frac{p_2}{2p_2 - p_1}$. We can still use the ideas of Theorem 4.1 to improve this factor. Notice, however, that although all of our results so far are independent of $\alpha(\cdot, \cdot)$, now the improvement will depend on the edge constraints. As in Algorithm 1, we can define a bipartite graph by using a restricted subset of the edges of G . In analogy to the set E' in section 4.1, we let $E' = \{(u, v) \in E : \text{val}(u) = p_2, \text{val}(v) = p_1, \text{ and } \alpha(u, v) < p_2 - p_1\}$, and $\alpha = \max_{(v_1, v_2) \in E'} \alpha(v_2, v_1)$. We have the following.

Theorem 4.6. *When $P = \{p_1, p_2\}$ there is a polynomial time ρ -approximation algorithm for the Inequity Aversion Pricing problem, where $\rho = \frac{p_2^2}{2p_2^2 - p_1 p_2 - (p_2 - p_1) \min(p_1, p_2 - p_1 - \alpha)}$. Furthermore, this ratio is tight.*

Notice that Theorem 4.6 yields a 0.8-approximation when $P = \{1, 2\}$. Finally, based on the improved approximation for two prices, we can get an analog of Theorem 4.3 for any number of distinct prices. Given an instance I , let I' be the new instance where for every $v \in V$, $val'(v) = \min\{val(v), p_2\}$, while the constraints remain the same.

Theorem 4.7. *Let $P = \{p_1, p_2, \dots, p_k\}$, and suppose that on instance I' (described above) the algorithm implied by Theorem 4.6 gives a $\frac{1}{p_2-x}$ -approximate solution. Then, we can get a $\frac{1}{p_k-x}$ -approximate solution for the original instance of the Inequity Aversion Pricing problem in polynomial time.*

In the Appendix, for Theorems 4.6 and 4.7 we provide the corresponding algorithms, along with proof sketches that highlight the differences between these proofs and the proofs of Theorems 4.1 and 4.3 respectively.

Table 1 summarizes approximation ratios obtained by the optimal single price solution, Algorithm 2, as well as the algorithm implied by Theorem 4.7 for different sets of prices.

P	$\{1, 2\}$	$\{1, 2, 3\}$	$\{1, \dots, 100\}$	$\{10, 20, 25\}$	$\{3, 6, 10, 11\}$
$1/H_k$	0.667	0.545	0.193	0.545	0.48
Alg. 2, general α	0.8	0.631	0.202	–	–
Thm. 4.7, general $\alpha \parallel \alpha = 0$	0.8	0.631	0.202	0.597 \parallel 0.689	0.524 \parallel 0.574

Table 1: Examples of obtained approximation ratios.

5 Hardness for Single Value Revenue Functions

In [2] there is an $n^{1-\varepsilon}$ inapproximability result for Inequity Aversion Pricing, but for general revenue functions and $\alpha(u, v) = 1$ for every edge. An NP-hardness proof is also given for these edge constraints when single value and constant revenue functions are allowed. The NP-hardness of Inequity Aversion Pricing as we study it here, i.e., allowing only single value revenue functions, was left as an open question. We resolve this question by proving that the problem remains NP-complete even if we restrict the revenue functions to be single value. Our reduction implies that the result holds even when the price differences are allowed to be close to the maximum possible price k . Further, when $\alpha(u, v) = 0$ for every edge, we are able to show APX-hardness.

The reduction, below, is from the decision version of *3-Terminal Node Cut*: Given a graph $G(V, E)$, a set $S = \{v_1, v_2, v_3\} \subseteq V$, and an integer q , is there a subset of q vertices that can be deleted, so that any two vertices of S are in different connected components of the resulting graph? The NP-completeness of the weighted version of 3-Terminal Node Cut is discussed in [8], while the APX-hardness of the unweighted version we use here is discussed in [13]; in either case no explicit proof is given. The NP-completeness result we need follows from Theorem 5.4 as well.

Theorem 5.1. *Let $\varepsilon > 0$ be any small constant. The decision version of Inequity Aversion Pricing (for single value revenue functions) is NP-complete even when $\alpha(u, v)$ is as large as $k^{1-\varepsilon}$ for all $(u, v) \in E(G)$, where k is the maximum possible price.*

Proof. It is immediate that the problem is in NP. To facilitate the presentation, we prove the NP-hardness when $\alpha(\cdot, \cdot)$ is upper bounded by $k^{1/3}/3$. As discussed at the end of the proof, the reduction can be easily adjusted when the upper bound of $\alpha(\cdot, \cdot)$ is $k^{1-\varepsilon}$, for constant ε .

Let us consider an instance of 3-Terminal Node Cut, i.e., a graph $G(V, E)$, with $|V(G)| = n$, a set $S = \{v_1, v_2, v_3\}$ of non adjacent vertices of G , and an integer q . We may assume that $q \leq n - 3$, otherwise the question is trivial. Next we give a construction of an appropriate instance for Inequity Aversion Pricing.

Let H be the graph obtained from G as follows. We replace every vertex $v \in S$ by n^3 vertices, where each such vertex has the same neighbors as v , i.e., if u_v is a vertex in the bundle of vertices replacing v , then for every edge $(v, x) \in E(G)$ we add the edge (u_v, x) to $E(H)$. For any $v \in S$, we call such a set of vertices in H a v -bundle. The set of prices is $\{\perp, 1, 2, \dots, k\}$, where $k = n^3 + n^2$. Finally, for any $(u, v) \in E(H)$ we set $\alpha(u, v)$ and $\alpha(v, u)$ arbitrarily, as long as they are at most $k^{1/3}/3$. Note that $|V(H)| = n - 3 + 3n^3$, and $|E(H)| \leq |E(G)| + 3(n - 1)n^3 \leq 3n^4$.

Next we define the single value revenue functions for the vertices of H . For every $v \in V(G) \setminus S$, let $val(v) = n^3 + n^2$, and for every $v_i \in S$, let $val(u_{v_i}) = n^3 + \frac{i-1}{2}n^2$ for all u_{v_i} in the v_i -bundle. We show below that G has a subset of at most q vertices that separate all the vertices of S , if and only if there is a feasible choice of prices for the vertices of H that gives revenue at least R_q , where $R_q = (n - 3 - q)n^3 + \sum_{i=1}^3 n^3(n^3 + \frac{i-1}{2}n^2)$.

One direction is easy. Let A be a subset of at most q vertices of G that separate the three vertices of S . For all $v \in A$ we put a discontinuity on the corresponding v in H . If we think of these vertices as removed from H , this creates several connected components. For any other vertex $u \in V(H)$, if u is in the same component as some v_i -bundle (or itself is one of the vertices of the v_i -bundle), set its price to $n^3 + \frac{i-1}{2}n^2$, otherwise set its price to $n^3 + n^2$. Notice that any vertex without a discontinuity produces revenue at least n^3 , while any vertex u_{v_i} in a v_i -bundle with $v_i \in S$ produces revenue exactly $n^3 + \frac{i-1}{2}n^2$. Now, it is straightforward to check that this price vector \mathbf{p} is feasible and gives enough revenue: $R(\mathbf{p}) = \sum_{u \in V(H)} R(u) \geq (n - 3 - q)n^3 + \sum_{i=1}^3 n^3(n^3 + \frac{i-1}{2}n^2) = R_q$.

For the opposite direction we begin with a couple of observations. Assume that there is a price vector \mathbf{p}_* that gives revenue at least R_q . We claim that \mathbf{p}_* can have only a few discontinuities.

Claim 5.2. *There is no feasible price vector \mathbf{p} with $R(\mathbf{p}) \geq R_q$ and more than q discontinuities.*

Proof of Claim 5.2. Let \mathbf{p} be a feasible price vector with at least $q + 1$ discontinuities. Notice that any vertex without a discontinuity produces revenue at most $n^3 + n^2$ and, in particular, any vertex u_{v_i} in a v_i -bundle with $v_i \in S$ produces revenue at most $n^3 + \frac{i-1}{2}n^2$. The maximum possible revenue for \mathbf{p} is

$$\begin{aligned} R(\mathbf{p}) &\leq (n - 3)(n^3 + n^2) + \sum_{i=1}^3 n^3 \left(n^3 + \frac{i-1}{2}n^2 \right) - (q + 1)n^3 \\ &= R_q + (n - 3)n^2 - n^3 < R_q, \end{aligned}$$

thus proving the claim. \triangleleft

One immediate implication of Claim 5.2 is that for any $v \in S$ not every vertex in the v -bundle has price \perp . This holds because the v -bundle has n^3 vertices and only $q \leq n - 3$ of them can get \perp . This is crucial, because if we think of the vertices with price \perp as removed from H , then no two vertices are separated because of discontinuities in the v -bundles. In particular, we can completely ignore those discontinuities with respect to connectivity.

Let $D_{\mathbf{p}} = \{v \in V(G) \setminus S \mid \mathbf{p}_v = \perp\}$, i.e., $D_{\mathbf{p}}$ is the set of non terminal vertices in G that their corresponding vertices in H have discontinuities in \mathbf{p} . So far, by Claim 5.2, we have that $|D_{\mathbf{p}_*}| \leq q$. What is left to be shown is that these discontinuities separate the v -bundles, for any $v \in S$.

Claim 5.3. *There is no feasible price vector \mathbf{p} such that $R(\mathbf{p}) \geq R_q$, and for some $v_i, v_j \in S$ vertices from both the v_i -bundle and the v_j -bundle are in the same connected component of the graph $H' = H - \{v \in V(H) \mid v \text{ is not in a bundle and } \mathbf{p}_v = \perp\}$.*

Proof of Claim 5.3. Let \mathbf{p} be a feasible price vector and assume that there exist $v_i, v_j \in S$ such that vertices from both the v_i -bundle and the v_j -bundle are in the same component of H' . First notice that all the vertices in the v_i -bundle and the v_j -bundle are in the same component, since vertices in a bundle share the same neighbors. We are going to upper bound the maximum possible revenue for such a price vector. W.l.o.g., assume $i < j$. If all the vertices in the v_i -bundle are assigned prices in $\{\perp, n^3 + \frac{i-1}{2}n^2 + 1, \dots, k\}$, then they contribute 0 to the total revenue. On the other hand, if there is some vertex in the v_i -bundle with price at most $n^3 + \frac{i-1}{2}n^2$, then by the feasibility of \mathbf{p} we have that any vertex in the v_j -bundle has its revenue upper bounded by $n^3 + \frac{i-1}{2}n^2 + \frac{k^{1/3}}{3}n$. To see the latter, notice that if any two vertices from two distinct bundles are connected by a path, then this path has length at most n (like it would in G) and therefore their prices can differ by $\frac{k^{1/3}}{3}n$ at most. We conclude that the loss, compared to the sum of the maximum revenues per vertex, is lower bounded by either $n^3(n^3 + \frac{i-1}{2}n^2)$ or $n^3(\frac{j-i}{2}n^2 - \frac{k^{1/3}}{3}n)$ and therefore by $n^3(\frac{1}{2}n^2 - \frac{k^{1/3}}{3}n)$. For $n \geq 10$, we have

$$n^3 \left(\frac{n^2}{2} - \frac{(n^3 + n^2)^{1/3}n}{3} \right) \geq n^3 \left(\frac{n^2}{2} - \frac{1.1^{1/3}n^2}{3} \right) > 0.15n^2n^3 > (n-2)n^3 \geq (q+1)n^3,$$

and we get $R(\mathbf{p}) < R_q$ in exactly the same way as in the proof of Claim 5.2. \triangleleft

We conclude that $D_{\mathbf{p}_*}$ is a set of at most q vertices of G that separate all the vertices of S . This completes the proof for the case where $\alpha(\cdot, \cdot)$ is upper bounded by $k^{1/3}/3$.

The above reduction, however, generalizes for $\alpha(\cdot, \cdot)$ upper bounded by $k^{1-\varepsilon}$ for any positive constant ε . Let $c \in \mathbb{N}$ with $c > 4/\varepsilon$. If we multiply by n^c all the relevant quantities, i.e., the size of the bundles, k , R_q , and $val(v)$ for all $v \in V(H)$, then the reduction is identical up to the last part of the proof of Claim 5.3. Now, the loss is lower bounded by $n^{c+3}(n^{c+2}/2 - nk^{1-\varepsilon})$ and it suffices for this quantity to be at least $(q+1)n^{c+3}$ for things to work out. So, we need $n^{c+2}/2 - nk^{1-\varepsilon} \geq n-2$ (since $n-2 \geq q+1$), and it is only a matter of simple calculations to check that this holds. \square

For the special case where all the differences are 0, we show that the problem is APX-hard. In doing so, we prove that 3-Terminal Node Cut is MAX SNP-hard, and thus APX-hard. As noted already, MAX SNP-hardness of 3-Terminal Node Cut is discussed—but not explicitly proved—in [13]. Here, having this reduction is crucial, and we have therefore obtained an explicit construction, since eventually we need to show that 3-Terminal Node Cut restricted in a specific set of instances is MAX SNP-hard (Corollary 5.5).

Theorem 5.4. *Multi-Terminal Node Cut is MAX SNP-hard even for 3 terminals and all the weights equal to 1.*

Proof. We prove the result for 3 terminals. The extension to more follows immediately. Proofs of MAX SNP-hardness involve linear reductions. Let A and B be two optimization problems. We say that A linearly reduces to B if there are two polynomial time computable functions f and g and constants $c_\alpha, c_\beta > 0$ such that

- Given an instance a of A , f produces an instance $b = f(a)$ of B such that $\text{OPT}_B(b) \leq c_\alpha \text{OPT}_A(a)$, and
- Given $a, b = f(a)$, and any solution y of b , g produces a solution x of a such that $|\text{cost}_A(x) - \text{OPT}_A(a)| \leq c_\beta |\text{cost}_B(y) - \text{OPT}_B(b)|$.

The reduction is from the unweighted version of *3-Terminal Cut*: Given a graph $G(V, E)$ and a set $S = \{v_1, v_2, v_3\} \subseteq V$, find a minimum cardinality set of edges that can be deleted, so that any two vertices of S are in different connected components of the resulting graph. 3-Terminal Cut was shown to be MAX SNP-hard in [9] even when all the weights equal to 1, which is essentially the unweighted version defined above.

Consider an instance of 3-Terminal Cut, i.e., a graph $G(V, E)$ with $|V(G)| = n$ and a set of non adjacent terminals $S = \{v_1, v_2, v_3\}$. We first describe the function f in the definition of the linear reduction. Let H be the graph obtained from G as follows:

1. Replace each edge e by a path of length two, the middle vertex of which we denote by v_e .
2. Replace every “old” vertex v by a v -bundle of $\deg_G(v) + 1$ vertices (see also the proof of Theorem 5.1), where each such vertex has the same neighbors as v in the graph constructed at step 1. That is, put an edge between u_v and v_e if u_v is a vertex in the v -bundle and e is incident to v .

Also, let $S' = \{u_1, u_2, u_3\}$, where u_i is an arbitrarily chosen vertex from the v_i -bundle. Define $f((G, S)) = (H, S')$. Clearly, f is polynomial time computable.

Next we define the function g in the definition of the linear reduction. Given a vertex cut Y in H that separates the vertices in S' , first we transform it to an appropriate vertex cut Y' that separates the vertices in S' and contains no vertices from any v -bundle.

1. While there is a whole v -bundle contained in the vertex cut, remove those vertices from the cut and add all of their neighbors instead.
2. While there is some vertex from a v -bundle in the cut, just remove this vertex from the cut.

Notice that in one iteration of step 1 the connectivity is not improved and the size of the vertex cut is reduced. The latter holds because $\deg_G(v) + 1$ vertices were removed from the cut and at most $\deg_G(v)$ were added. Similarly, in one iteration of step 2 the connectivity is not improved and the size of the vertex cut is reduced. Now the latter is obvious, but to see that the connectivity is not improved, notice that the removal of vertices in some v -bundle has an effect in connectivity only if the whole v -bundle is removed. Since in step 2 there are no v -bundles completely contained in the vertex cut (this was fixed in step 1), the vertices removed from the cut were not disconnecting anything to begin with. We conclude that Y' is indeed a vertex cut that separates the vertices in S' and moreover $|Y'| \leq |Y|$.

Now, that Y' contains only vertices outside the v -bundles, i.e., only vertices that correspond to edges of G , it is straightforward to define an edge cut in G that separates the vertices in S . Let $X = \{e \in E(G) \mid v_e \in Y'\}$, i.e., X is the set of edges in G that their corresponding vertices in H are in the vertex cut. Define $g((G, S), (H, S'), Y)$ to be equal to X ; clearly, g is polynomial time computable. It remains to be shown that X separates the vertices in S . Assume not; then there exists some $v_i - v_j$ path $p = (v_i, x_1, x_2, \dots, x_k, v_j)$ in $G - X$ for $v_i, v_j \in S$, with $i \neq j$. This, however, directly transforms to a $u_i - u_j$ path $p' = (u_i, v_{(v_i, x_1)}, x'_1, v_{(x_1, x_2)}, x'_2, \dots, x'_k, v_{(x_k, v_j)}, u_j)$ in $H - Y'$,

where x'_ℓ is an arbitrary vertex in the x_ℓ -bundle. This is a contradiction. Thus, X is a cut that separates the vertices in S .

Next, we prove that $\text{OPT}_{3\text{TNC}}(H) \leq \text{OPT}_{3\text{TC}}(G)$ (to improve readability we drop the subscripts). Notice that any cut A in G that separates the vertices of S gives the vertex cut $B = \{v_e \in V(H) \mid e \in A\}$ in H that separates the vertices of S' . Since $|B| = |A|$, and by taking $|A|$ to be an optimal cut, we have $\text{OPT}(H) \leq \text{OPT}(G)$. This also implies that $c_\alpha = 1$ works.

Finally, since $|X| = |Y'|$, we have $|X| - \text{OPT}(G) \leq |Y'| - \text{OPT}(H) \leq |Y| - \text{OPT}(H)$, i.e., $c_\beta = 1$ works. We conclude that the unweighted version of 3-Terminal Node Cut is MAX SNP-hard. \square

As proved in [18], APX is the closure of MAX SNP under PTAS reductions (introduced by [7]). Therefore, any MAX SNP-hard problem is also APX-hard. Let \mathcal{J} be the set of instances of 3-Terminal Node Cut that can be the result of the composition of the reduction of Theorem 5.4 with the linear reduction from Max Cut to 3-Terminal Cut, presented in [9]. The next corollary follows directly.

Corollary 5.5. *3-Terminal Node Cut is MAX SNP-hard, and thus APX-hard, even when restricted on instances in \mathcal{J} .*

Corollary 5.5 is a crucial step towards our goal, since instances in \mathcal{J} are guaranteed to have only “large” vertex cuts that separate the terminals.

Lemma 5.6. *Let $(G, S, q) \in \mathcal{J}$. Then, any feasible 3-Terminal Node Cut solution for (G, S, q) has size greater than $\frac{1}{14}|V(G)|$.*

Proof. Let G_0 be a graph with n_0 vertices and m_0 edges. The reduction of [9] adds 3 terminals and, furthermore, for each edge adds 4 new vertices and 102 new edges. In fact, each edge is replaced with the gadget shown in Figure 2 (Figure 11 of [9]), where s_1, s_2, s_3 are identified with the terminals and x, y with the endpoints of the edge. Then, each of the 12 edges with weight 4 is replaced by 4 paths of length 2. The resulting graph G_1 , has $n_1 = n_0 + 3 + 52m_0$ vertices and $m_1 = 102m_0$ edges.

Our reduction adds 1 new vertex for each edge, and then replaces each one of the old vertices with $\deg_{G_1}(v) + 1$ new vertices. The number of vertices of the resulting graph G_2 is $n_2 = \sum_{v \in V(G_1)} (\deg_{G_1}(v) + 1) + m_1 = n_1 + 3m_1 = n_0 + 3 + 358m_0 < 378m_0$.

By the proof of Theorem 3 in [9], we have that any cut in G_1 that separates the 3 terminals has size at least $27m_0$. Using g from our reduction, however, we can transform a vertex cut that separates the 3 terminals in G_2 into a cut of the same cardinality that separates the 3 terminals in G_1 . Thus, any vertex cut that separates the 3 terminals in G_2 has size at least $27m_0$. To complete the proof, notice that $27m_0 > 27n_2/378 = n_2/14$. \square

Theorem 5.7. *Inequity Aversion Pricing (for single value revenue functions) is APX-hard when $\alpha(e) = 0$ for all $e \in E(G)$.*

Proof. We use a PTAS reduction to prove the APX-hardness. Let A and B be two NPO problems. Here assume that A is a minimization and B is a maximization problem. We say that A is PTAS-reducible to B if there exist three computable functions f , g , and c such that

- For any instance x of A and any $r > 1$, $f(x, r)$ is an instance of B computable in time polynomial in $|x|$.

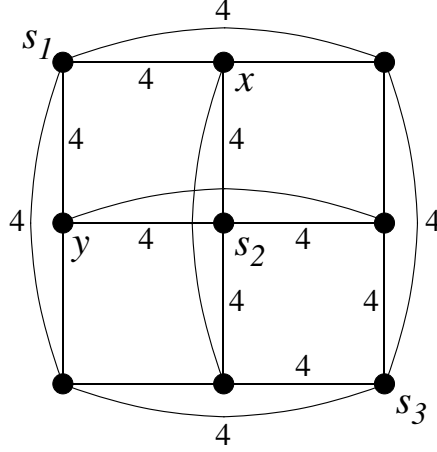


Figure 2: The gadget that “replaces” every edge in the linear reduction of from Max Cut to 3-Terminal Cut [9].

- For any instance x of A , any $r > 1$, and any feasible solution y of $f(x, r)$, $g(x, y, r)$ is a feasible solution of A computable in time polynomial in both $|x|$ and $|y|$.
- $c : (1, \infty) \rightarrow (0, 1)$
- For any instance x of A , any $r > 1$, and any feasible solution y of $f(x, r)$,

$$\text{cost}_B(y) \geq c(r) \cdot \text{OPT}_B(f(x, r)) \text{ implies } \text{cost}_A(g(x, y, r)) \leq r \cdot \text{OPT}_A(x).$$

The reduction is from the restriction of 3-Terminal Node Cut on \mathcal{J} . It is similar to the reduction in the proof of Theorem 5.1, but here all the parameters are carefully tuned. Consider an instance of 3-Terminal Node Cut, i.e., a graph $G(V, E)$ with $|V(G)| = n$, a set of non adjacent terminals $S = \{v_1, v_2, v_3\}$, and an integer q , such that $(G, S, q) \in \mathcal{J}$. We describe the function f in the definition of a PTAS reduction.

For $r > 1$, let $\varepsilon = \min\{0.5, r - 1\}$ and $t = \lceil \frac{42}{\varepsilon} \rceil$. Also, let $H = f(G, r)$ be the graph obtained from G by replacing every vertex $v \in S$ by a v -bundle of $4tn$ vertices, each such vertex having the same neighbors as v . The set of prices is $\{\perp, 1, 2, \dots, t\}$. To define the single value revenue functions, for every $v \in V(G) \setminus S$, let $\text{val}(v) = t$, and for every $v_i \in S$, let $\text{val}(u_{v_i}) = t + i - 3$ for all u_{v_i} in the v_i -bundle. We define $f((G, S, q), r)$ to be the above instance. Clearly, f is computable in polynomial time in n .

Next we define the function g in the definition of a PTAS reduction. Given a feasible price vector \mathbf{p} for H , first we transform it to an appropriate feasible price vector \mathbf{p}' .

1. While there is a whole v -bundle only with discontinuities, set price $\text{val}(u_v)$ to all the vertices u_v in this v -bundle and \perp to all of their neighbors.
2. Consider the graph after we remove all the vertices with price \perp . While there are i, j (assume $i < j$) such that vertices from both the v_i -bundle and the v_j -bundle are in the same connected component:
 - If all the vertices in the v_i -bundle are assigned prices in $\{\perp, \text{val}(u_{v_i}) + 1, \dots, t\}$, then set price $\text{val}(u_{v_i})$ to all the vertices in the v_i -bundle and \perp to all of their neighbors.

- Otherwise, set price $val(u_{v_j})$ to all the vertices in the v_j -bundle and \perp to all of their neighbors.

Then, we use this price vector \mathbf{p}' in order to define a solution D to the 3-Terminal Node Cut instance. Let $D = \{v \in V(G) \setminus S \mid \mathbf{p}'_v = \perp\}$, i.e., D is the set of non terminal vertices in G that their corresponding vertices in H have discontinuities. Again, it is straightforward to see that computing $g((G, S, q), \mathbf{p}, r)$ takes polynomial time in n .

It remains to determine the function c in the definition of the reduction; for any $r \in (1, \infty)$ let $c(r) = 1 - \frac{1}{20t^2}$. We need to show that

$$R(\mathbf{p}) \geq c(r) \cdot \text{OPT}(H) \implies \text{cost}(D) \leq r \cdot \text{OPT}(G).$$

Claim 5.8. *If $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$, then $\mathbf{p}' = \mathbf{p}$, i.e., there is no v -bundle only with discontinuities, and every v -bundle is in a different connected component.*

Proof of Claim 5.8. For the first part, assume that there is a v -bundle, where every single vertex gets price \perp . We get the following upper bound for $R(\mathbf{p})$:

$$R(\mathbf{p}) \leq (n-3)t + \sum_{i=2}^3 4tn(t+i-3) \leq 8t^2n - 3tn - 3t < 8t^2n.$$

On the other hand, there is a feasible price vector that sets all the prices equal to $t-2$, and this way we get a lower bound on $\text{OPT}(H)$.

$$\text{OPT}(H) \geq (3 \cdot 4tn + n - 3)(t-2) = 12t^2n - 23tn - 2n - 3t + 6 > 12t^2n - 28tn.$$

Notice that, since $\varepsilon \leq 0.5$, we have $t \geq 84$, and therefore $c(r) > 0.99$. So, we have $\frac{R(\mathbf{p})}{\text{OPT}(H)} < \frac{8t}{12t-28} < 0.8 < c(r)$, which is a contradiction.

For the second part, assume that there are two v -bundles in the same component. To arrive at a contradiction, it suffices to show that there exists a feasible price vector \mathbf{p}'' , such that $R(\mathbf{p}) < c(r)R(\mathbf{p}'')$ and therefore $R(\mathbf{p}) < c(r)\text{OPT}(H)$. Let \mathbf{p}'' be the price vector obtained after just one iteration of step 2 in the description of g . Assuming that we are talking about the v_i -bundle and the v_j -bundle, with $i < j$, the gain in revenue is at least $4tn(val(u_{v_j}) - val(u_{v_i})) \geq 4tn$ (see also the proof of Claim 5.3 in the proof of Theorem 5.1). On the other hand, the loss in revenue is upper bounded by $(n-3)(t+i-3) \leq tn$. So, $R(\mathbf{p}'') \geq R(\mathbf{p}) + 3tn$. Suppose $R(\mathbf{p}) \geq c(r)R(\mathbf{p}'')$. Then it is a matter of simple calculations to see that

$$R(\mathbf{p}) \geq c(r)(R(\mathbf{p}) + 3tn) \implies R(\mathbf{p}) \geq 60t^3n - 3tn > 57t^3n.$$

An obvious upper bound for $R(\mathbf{p})$ however, is to say that each vertex produces revenue at most t , i.e., $R(\mathbf{p}) \leq (12tn + n - 3)t < 13tn$. Combining the two, we get the contradiction $R(\mathbf{p}) > 57t^3n > 13tn > R(\mathbf{p})$. We conclude that $R(\mathbf{p}) < c(r)R(\mathbf{p}'')$, that leads to the contradiction $R(\mathbf{p}) < c(r)\text{OPT}(H)$. Hence, in the graph defined by removing the discontinuities of \mathbf{p} from H , every v -bundle is in a different connected component. \triangleleft

Claim 5.9. *If $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$, then \mathbf{p} has less than $(1 + \varepsilon)\text{OPT}(G)$ discontinuities.*

Proof of Claim 5.9. Let \mathbf{p} be a feasible price vector with $R(\mathbf{p}) \geq c(r)\text{OPT}(H)$ and assume that \mathbf{p} has at least $(1 + \varepsilon)\text{OPT}(G)$ discontinuities. Also, consider the feasible price vector \mathbf{p}^* induced by an optimal cut in G , i.e., the price vector that sets \perp in every vertex that has a corresponding vertex removed by the cut in G and then uses optimal single price in each “connected component”. To get a contradiction, we show that $R(\mathbf{p}) < c(r)R(\mathbf{p}^*)$ and therefore $R(\mathbf{p}) < c(r)\text{OPT}(H)$. To obtain a lower bound on $R(\mathbf{p}^*)$, notice that any vertex without a discontinuity produces revenue at least $t - 2$, while any vertex u_{v_i} in a v_i -bundle produces revenue exactly $t + i - 3$. So,

$$R(\mathbf{p}^*) \geq (n - 3 - \text{OPT}(G))(t - 2) + \sum_{i=1}^3 4tn(t + i - 3).$$

To get an upper bound for $R(\mathbf{p})$, notice that each vertex without a discontinuity produces revenue at least $t - 2$ and at most t , while any vertex u_{v_i} in a v_i -bundle produces revenue exactly $t + i - 3$, i.e.,

$$R(\mathbf{p}) \leq (n - 3)t - (1 + \varepsilon)\text{OPT}(G)(t - 2) + \sum_{i=1}^3 4tn(t + i - 3).$$

We consider the difference $R(\mathbf{p}) - c(r)R(\mathbf{p}^*)$, and show it is negative. Recall that Lemma 5.6 implies that $\text{OPT}(G) \geq n/14$.

$$\begin{aligned} R(\mathbf{p}) - c(r)R(\mathbf{p}^*) &\leq \frac{1}{20t^2} \left((n - 3 - \text{OPT}(G))(t - 2) + \sum_{i=1}^3 4tn(t + i - 3) \right) \\ &\quad + 2(n - 3) - \varepsilon \text{OPT}(G)(t - 2) \\ &< \frac{1}{20t^2} (nt + 12t^2n) + 2n - \varepsilon \frac{1}{14}n \left(\frac{42}{\varepsilon} - 2 \right) \\ &< \frac{13n}{20} + 2n - 2.9n < -0.25n < 0, \end{aligned}$$

which leads to contradiction. Thus, \mathbf{p} has less than $(1 + \varepsilon)\text{OPT}(G)$ discontinuities. \triangleleft

By combining Claim 5.8, Claim 5.9, and the fact that $1 + \varepsilon \leq r$, we directly get that a $c(r)$ -approximate solution for H gives an r -approximate solution for G , thus concluding the proof. \square

Remark 2. The maximum price k in the instance constructed in the proof of Theorem 5.7 does not depend on the size of the problem. Given that there is some constant ρ beyond which it is hard to approximate 3-Terminal Node Cut, this means that there exists some constant k^* for which Inequity Aversion Pricing does not have a PTAS. Note that for such a k^* we do have a constant factor approximation, with factor $H_{k^*}^{-1}$.

6 A Generalization to Multi-Demand Users

So far, we have always assumed that each node has demand for only one copy of the product. A natural generalization is to consider multi-demand users who are interested in receiving a certain number of copies if the price is affordable. For example, someone might want to buy either a certain number of licenses of a video game (because she wants to play the game with her friends) or no license at all. This would correspond to a type of inelastic multi-unit demand in the terminology of auctions. Assume again that there is enough supply of copies to satisfy all the demand, if necessary. Then, there is a natural way to generalize single value revenue functions to capture such simple scenarios.

A revenue function $R_v(\cdot)$ is called a *multi-demand single value revenue function* if there exist an integer s_v (the number of copies demanded) and a value $val(v)$ such that:

$$R_v(p_v) = \begin{cases} s_v p_v & \text{if } val(v) \geq p_v \\ 0 & \text{if } val(v) < p_v \end{cases}.$$

The intuition here is the same as for the single value revenue functions.

The objective now is again the same. Given a multi-demand single value revenue function for each node, find a feasible price vector \mathbf{p} that maximizes the total revenue. We call this problem *Multi-Demand Inequity Aversion Pricing*. As this is a generalization of Inequity Aversion Pricing, it is immediate that any negative result for the latter yields the same negative result for the former. In particular, by Theorems 5.1 and 5.7, Multi-Demand Inequity Aversion Pricing is NP-hard and APX-hard for the corresponding edge constraints.

Quite surprisingly, we also prove that when for each user the number of demanded copies is polynomially bounded, there is a strict reduction from Multi-Demand Inequity Aversion Pricing to Inequity Aversion Pricing. This directly implies that any approximation factor achieved for the latter is also achieved for the former. Therefore, we establish that the two problems are equivalent in terms of approximability. Note that the theorem holds for general edge constraints.

Theorem 6.1. *Let q be any polynomial. There exists a strict reduction from Multi-Demand Inequity Aversion Pricing with demands bounded by $q(n)$ to Inequity Aversion Pricing.*

Proof. Suppose we are given an instance I of Multi-Demand Inequity Aversion Pricing, i.e., a graph $G(V, E)$, an edge restriction function $\alpha(\cdot, \cdot)$, and for each node v her valuation $val(v)$ and her demand s_v . We are going to construct an equivalent instance I' of Inequity Aversion Pricing. The reduction creates s_v copies of v for each $v \in V$ and connects them to each other to create a clique K_{s_v} . Edges inside such a clique have $\alpha = 0$. For every edge $(u, v) \in E$ all the edges between the vertices of the u -clique and the v -clique are added with the same restrictions as the original edge. Let $G' = (V', E')$ be the resulting graph. If $s_{\max} = \max_{v \in V} s_v$ then we have $|V'| \leq ns_{\max}$ and $|E'| \leq (n + m)s_{\max}^2$.

We use OPT' and OPT to denote the optimal revenue of this instance and of the original, respectively. Our goal is to show that for any price vector \mathbf{p}' for I' we can efficiently find a feasible price vector \mathbf{p} for I with such that $\frac{R(\mathbf{p})}{OPT} \geq \frac{R'(\mathbf{p}')}{OPT'}$. We begin by proving that $OPT = OPT'$.

Claim 6.2. *An optimal price vector \mathbf{p}' for I' sets the same price for all vertices inside each v -clique.*

Proof of Claim 6.2. Note that $\alpha = 0$ inside each v -clique, so all these vertices have the same common price p'_v or \perp . If there are x, y in a v -clique such that $p'_x \neq \perp \wedge p'_y = \perp$ then by setting $p'_y = p'_x$ we obtain a new feasible price vector for I' that gives greater revenue than \mathbf{p}' , which contradicts its optimality. \triangleleft

By Claim 6.2, we directly obtain a feasible solution for I with revenue OPT by setting p_v equal to the common price from the v -clique. Therefore, $OPT \geq OPT'$.

On the other hand, each feasible price vector \mathbf{p} for I can be adopted as a feasible price vector \mathbf{p}' for I' with the same revenue. To see that, just set the same price $p'_{u_v} = p_v$ for each copy u_v of v in the v -clique of G' . All edge constraints are satisfied, so the solution is feasible, and it clearly gives the same revenue. By taking \mathbf{p} to be an optimal price vector for I , the above implies that $OPT' \geq OPT$. We conclude that $OPT = OPT'$.

Finally, we need the following.

Claim 6.3. *Each feasible price vector \mathbf{p}' for I' can be transformed into a feasible price vector \mathbf{p} for I with at least the same revenue.*

Proof of Claim 6.3. For each $u \in V$, if V_u is the set of vertices in the u -clique of G' , define $u^* = \arg\max_{x \in V_u} p'_x$. Then, set $p_u = p'_{u^*}$. Such a \mathbf{p} is feasible for I because $\forall (u, v) \in E$, $\alpha(u, v) = \alpha(u^*, v^*)$, where v^* is any vertex in V_v , and the constraint $\alpha(u^*, v^*)$ is already satisfied by \mathbf{p}' . It is straightforward that $R(\mathbf{p}) \geq R'(\mathbf{p}')$. \triangleleft

For the price vector described in the proof of Claim 6.3, we have

$$\frac{R(\mathbf{p})}{\text{OPT}} \geq \frac{R'(\mathbf{p}')}{\text{OPT}} = \frac{R'(\mathbf{p}')}{\text{OPT}'},$$

which completes the proof. \square

It would be interesting to determine whether the hardness of the problem changes when the demands are not polynomially bounded, although such functions are not very realistic in our setting. Notice, however, that even in that case it is not hard to obtain a $\frac{1}{H_k}$ -approximation in polynomial time by using the best single-price solution. In fact, we still have a $\frac{1}{H_r}$ -approximation, where $r = \min\{n, v_{\max}\}$.

Concluding remarks

We studied a revenue maximization problem under inequity aversion for the natural class of single-value revenue functions. Apart from establishing the first hardness results for this class, we also derived approximation algorithms based on combinatorial and graph-theoretic tools, which improve the state of the art when the set of available prices is small. We find this to be a realistic setting as special price offers are usually derived by specific discount and promotion policies. Several questions still remain open. Even for $k = 2$ it is not known if the problem is NP-hard, or whether we can have better approximation ratios. Clearly, it would also be very interesting to resolve the approximability for general k , i.e., can we have a better than $O(1/H_k)$ -approximation for large k ? Exploring further models of negative externalities is another attractive direction that has not been given as much attention as the case of positive externalities.

Acknowledgements

This research was supported by National Science Centre, Poland, 2015/17/N/ST6/03684. It was also supported by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES.

References

- [1] H. Akhlaghpour, M. Ghodsi, N. Haghpanah, H. Mahini, V. S. Mirrokni, and A. Nikzad. Optimal iterative pricing over social networks (extended abstract). In *Proceedings of the 6th Workshop on Internet and Network Economics, WINE 2010*, pages 415–423, 2010.

- [2] N. Alon, Y. Mansour, and M. Tennenholtz. Differential pricing with inequity aversion in social networks. In *ACM Conference on Economics and Computation, EC 2013*, pages 9–24, 2013.
- [3] G. Amanatidis, E. Markakis, and K. Sornat. Inequity aversion pricing over social networks: Approximation algorithms and hardness results. In *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016*, 2016.
- [4] S. Bhattacharya, J. Kulkarni, K. Munagala, and X. Xu. On allocations with negative externalities. In *Proceedings of the 7th Workshop on Internet and Network Economics, WINE 2011*, pages 25–36, 2011.
- [5] G. E. Bolton and A. Ockenfels. A theory of equity, reciprocity and competition. *American Economic Review*, 100:166–193, 2000.
- [6] Z. Cao, X. Chen, X. Hu, and C. Wang. Pricing in social networks with negative externalities. In *Proceedings of the 4th International Conference on Computational Social Networks, CSoNet 2015*, pages 14–25, 2015.
- [7] P. Crescenzi and L. Trevisan. On approximation scheme preserving reducibility and its applications. *Theory Comput. Syst.*, 33(1):1–16, 2000.
- [8] W. H. Cunningham. The optimal multiterminal cut problem. In *Reliability Of Computer And Communication Networks, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, December 2-4, 1989*, pages 105–120, 1989.
- [9] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.
- [10] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM International Conference on Knowledge Discovery and Data Mining, KDD 2001*, pages 57–66, 2001.
- [11] E. Fehr and K. M. Schmidt. A theory of fairness, competition and co-operation. *Quarterly Journal of Economics*, 114:817–868, 1999.
- [12] D. Fotakis and P. Siminelakis. On the efficiency of influence-and-exploit strategies for revenue maximization under positive externalities. In *Proceedings of the 8th Workshop on Internet and Network Economics, WINE 2012*, pages 270–283, 2012.
- [13] N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.
- [14] A. Goldberg, J. Hartline, A. Karlin, M. Saks, and A. Wright. Competitive auctions. *Games and Economic Behavior*, 55(2):242–269, 2006.
- [15] N. Haghpahan, N. Immorlica, V. S. Mirrokni, and K. Munagala. Optimal auctions with positive network externalities. In *Proceedings of the 12th ACM Conference on Electronic Commerce, EC 2011*, pages 11–20, 2011.

- [16] J. Hartline, V. S. Mirrokni, and M. Sundararajan. Optimal marketing strategies over social networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 189–198, 2008.
- [17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, 2003.
- [18] S. Khanna, R. Motwani, M. Sudan, and U. V. Vazirani. On syntactic versus computational views of approximability. *SIAM J. Comput.*, 28(1):164–191, 1998.
- [19] J. Kleinberg. Cascading behavior in networks: Algorithmic and economic issues. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, *Algorithmic Game Theory*, chapter 24, pages 613–632. Cambridge University Press, Cambridge, 2007.
- [20] M. D. Plummer and L. Lovász. *Matching Theory*. North-Holland Mathematics Studies. Elsevier Science, 1986.

Appendix: Proof Sketches of Theorems 4.6 and 4.7

Proof Sketch of Theorem 4.6. As shown below, the algorithm for two general prices is very similar to Algorithm 1, but its analysis is more tedious.

Algorithm 3: An improved approximation when $P = \{p_1, p_2\}$, with $p_1 \leq p_2$

- 1 Given the graph $G = (V, E)$, construct the bipartite graph $G' = (V_1, V_2, E')$ with
 $V_i = \{v \in V : \text{val}(v) = p_i\}$ and $E' = \{(u, v) \in E : \text{val}(u) = p_2, \text{val}(v) = p_1, \text{ and } \alpha(u, v) < p_2 - p_1\}$
 - 2 Find a minimum vertex cover on G' , say $S \subseteq V$
 - 3 Set \perp to all vertices of S
 - 4 Set a price of p_1 to every $v \in V_1 \setminus S$ and a price of p_2 to every $v \in V_2 \setminus S$. Let R^* be the revenue obtained by this solution
 - 5 Compute the optimal single-price solution, as described in Section 3, with revenue R_{Sp}
 - 6 Return the solution that achieves $\max\{R^*, R_{\text{Sp}}\}$
-

We want to lower bound the approximation ratio γ of Algorithm 3. The proof is the same as the proof of Theorem 4.1 up to the point where we upper bound OPT. Here the analog of Claim 4.2 is $\text{OPT} \leq \text{OPT}' \leq \text{MAX} - |S| \cdot \min(p_1, p_2 - p_1 - \alpha)$ and can be proved using the same idea. To improve readability, we write r instead of $\min(p_1, p_2 - p_1 - \alpha)$.

Next we know that $\text{ALG} = \text{MAX} - \text{val}(S)$ and $\text{val}(S) \leq p_2 \cdot |S|$. Therefore, $|S| \geq \frac{1}{p_2} (\text{MAX} - \text{ALG})$. Combining with the upper bound for OPT, we obtain $\text{OPT} \leq \frac{p_2 - r}{p_2} \text{MAX} + \frac{r}{p_2} \text{ALG}$.

We can divide the interval $[0, \gamma \cdot \text{MAX}]$ the same way as in the proof of Theorem 4.1 and do a similar analysis. The two cases here are taken with respect to $i^* = \left\lceil \frac{(m+1)p_2 - mr}{p_2 - \gamma r} \right\rceil$. Moreover, for the case where $\text{ALG} < \frac{i^* - 1}{m} \cdot \gamma \cdot \text{MAX}$ we use the lower bound $\text{ALG} \geq \frac{p_2}{2p_2 - p_1} \cdot \text{MAX}$ (approximation ratio for single-price solution, implied by Theorem 4.5). At the end of the analysis we obtain

$$\gamma \geq \frac{p_2^2}{2p_2^2 - p_1 p_2 - (p_2 - p_1)r},$$

as desired.

The tightness of the approximation follows from the tightness of Theorem 4.1 for $p_1 = 1, p_2 = 2$ and $\alpha = 0$. \square

Proof Sketch of Theorem 4.7. The algorithm is very similar to Algorithm 2, so we only state the differences:

- In the definition of I' (step 1 of Alg. 2) we have $val'(v) = \min\{val(v), p_2\}, \forall v \in V$
- On instance I' we run Algorithm 3, instead of Algorithm 1 (step 2 of Alg. 2)

The proof is by induction on k . For $k = 2$, there is nothing to prove.

Now assume we have an instance I where $k > 2$. We use the notation of the proof of Theorem 4.3, but now R_k is the revenue extracted by setting price p_k at every node, and $V_k = \{v \in V : val(v) = p_k\}$.

Case (i): $|V_k| \geq \frac{1}{(p_k - x)p_k} \cdot \text{OPT}$. Then, $\frac{\text{ALG}}{\text{OPT}} \geq \frac{R_k}{\text{OPT}} = \frac{p_k \cdot |V_k|}{\text{OPT}} \geq \frac{1}{p_k - x}$.

Case (ii): $|V_k| < \frac{1}{(p_k - x)p_k} \cdot \text{OPT}$. Let I^* be an instance derived from I by setting $val^*(v) = \min\{val(v), p_{k-1}\}$.

By the inductive hypothesis we have $\text{ALG}^* \geq \frac{1}{p_{k-1} - x} \cdot \text{OPT}^*$. It is easy to see that $\text{ALG}^* \leq \text{ALG}$. Also, we can prove an analog of Claim 4.4, namely $\text{OPT}^* \geq \text{OPT} - (p_k - p_{k-1})|V_k|$. Putting everything together we have

$$\begin{aligned} \frac{\text{ALG}}{\text{OPT}} &\geq \frac{\text{ALG}^*}{\text{OPT}} \geq \frac{\frac{1}{p_{k-1} - x} \cdot \text{OPT}^*}{\text{OPT}} \geq \frac{\frac{1}{p_{k-1} - x} \cdot (\text{OPT} - (p_k - p_{k-1})|V_k|)}{\text{OPT}} \\ &\geq \frac{1}{p_{k-1} - x} \left(1 - \frac{\frac{p_k - p_{k-1}}{p_k(P_k - x)} \cdot \text{OPT}}{\text{OPT}} \right) = \frac{1}{p_{k-1} - x} \cdot \frac{\frac{p_k}{p_k - p_{k-1}} P_k - \frac{p_k}{p_k - p_{k-1}} x - 1}{\frac{p_k}{p_k - p_{k-1}} (P_k - x)} \\ &= \frac{1}{p_{k-1} - x} \cdot \frac{\frac{p_k}{p_k - p_{k-1}} (P_{k-1} - x)}{\frac{p_k}{p_k - p_{k-1}} (P_k - x)} = \frac{1}{p_k - x}, \end{aligned}$$

which concludes the proof. \square